

"Express Mail" mailing label number: EV 164034101 US

Date of Deposit: November 18, 2003

Attorney Docket No.15030US02

SYSTEM AND METHOD FOR EFFICIENTLY STORING MACROBLOCKS IN
SD-RAM

RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Application for Patent, Serial No., 60/484,512, entitled "SYSTEM AND METHOD FOR EFFICIENTLY STORING MACROBLOCKS IN SD-RAM", filed July 2, 2003, by Kumar, et. al..

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0002] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[0003] [Not Applicable]

BACKGROUND OF THE INVENTION

[0004] The MPEG-2 standard for encoding and compressing video data uses macroblocks for encoding individual frames from the video data. Each frame is associated with three matrices representing luminance and two chrominance (Cb and Cr) values. The Y matrix has an even number of rows and columns while the Cb and Cr matrices are one-half the size of the Y matrix in each direction (horizontal and vertical). Each matrix is further divided into 8 x 8 segments known as blocks. Each block from the chrominance matrices is associated with four blocks from the luminance matrix because the luminance matrix is twice the size of the chrominance

matrices in both directions. The blocks from the chrominance matrices and the associated four blocks from the luminance matrix together form a macroblock.

[0005] Each macroblock is compressed using a variety of algorithms taking advantage of both spatial and temporal redundancies. The macroblocks forming the frames of the video sequence are then packetized and multiplexed for transmission to a decoder that decodes the video sequence.

[0006] During the decoding of the video sequence, the frames of the video sequence are decoded and stored in frame buffers. The frame buffers store frames prior to display on a display device. Additionally, frame buffers also store reference frames that are used for decoding frames that are predicted therefrom.

[0007] The frames are decoded in units of macroblocks. After decoding a macroblock, the decoder writes the decoded macroblock into a frame buffer. As noted above, the macroblock includes a 16 x 16 luminance matrix, and two 8 x 8 chrominance matrices. In order to decode and display frames in real-time, it is advantageous if a simple addressing scheme is used. A particularly simple addressing scheme is based on integer powers of two. Each of the foregoing matrices have an integer power of two bytes (2^8 for the luminance matrix and 2^6 for the chrominance matrix). Accordingly, in a memory storing luminance matrices, the address of the xth luminance matrix is determined by offsetting the starting address with $x \cdot 2^8$. In a memory storing chrominance matrices, the address of the xth chrominance matrix is determined by offsetting the starting address with $x \cdot 2^6$.

[0008] Although the luminance and chrominance matrices have an integer power of two bytes, a macroblock comprises 384 bytes. To simplify the addressing scheme for macroblocks, the luminance, and two chrominance matrices are stored in separate, non-contiguous, frame buffer portions.

[0009] The foregoing simplifies the addressing scheme for the luminance and two chrominance matrices. However, during the writing of a macroblock, each matrix is written separately because the portions of the frame buffer written to are non-contiguous. As a result, a separate write transaction is required for writing each matrix. The foregoing requires more instructions and operations for writing a macroblocks.

[0010] Further limitations and disadvantages of conventional and traditional approaches will become apparent to one of skill in the art, through comparison of such systems with the present invention as set forth in the remainder of the present application with reference to the drawings.

SUMMARY OF THE INVENTION

[0011] A system, method, and apparatus for efficiently storing macroblocks in SD-RAM are presented herein.

[0012] In one embodiment, there is presented a method for storing macroblocks in a memory. The macroblocks are stored in memory by decoding a macroblock, and executing an instruction. The instruction causes the macroblock to be written in memory.

[0013] In another embodiment, there is presented a method for storing macroblocks in a memory. The macroblocks are stored in memory by decoding five macroblocks and executing an instruction. The instruction causes writing the five macroblocks to the memory.

[0014] In another embodiment, there is presented a circuit for storing macroblocks. The circuit comprises a decoder and a computer readable medium. The decoder decodes the macroblocks. The computer readable medium stores an executable instruction. The executable instruction causes writing of the macroblock to the memory.

[0015] In another embodiment, there is presented a circuit for storing macroblocks comprising a decoder and a computer readable medium. The decoder decodes five macroblocks. The computer readable medium stores an executable instruction. The instruction causes writing of the five macroblocks to the memory.

[0016] These and other advantages and novel features of the present invention, as well as details illustrated embodiments thereof, will be more fully understood from the following description and drawings.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[0017] **FIGURE 1** is a block diagram describing the encoding of video data in accordance with the MPEG-2 specification

[0018] **FIGURE 2** is a block diagram of an exemplary decoder in accordance with an embodiment of the present invention;

[0019] **FIGURE 3** is a block diagram of a frame buffer storing macroblocks in accordance with an embodiment of the present invention; and

[0020] **FIGURE 4** is a block diagram of a frame buffer storing macroblocks in accordance with another embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0021] Referring now to **FIGURE 1** there is illustrated a block diagram of the MPEG-2 video stream hierarchy. A video sequence 105 includes a number of groups 302, wherein each group 302 comprises an encoded representation of a series of pictures 305.

[0022] Each picture 305 is associated with three matrices representing luminance (Y) 305a and two chrominance (Cb and Cr) values, 305b, 305c. The Y matrix 305a has an even number of rows and columns while the Cb and Cr matrices 305b, 305c are one-half the size of the Y matrix in each direction (horizontal and vertical). Each matrix 305a, 305b, 305c is further divided into 8 x 8 segments known as blocks 310. Each block 310b, 310c from the chrominance matrices 305b, 305c is associated with four blocks 310a from the luminance matrix 305a because the luminance matrix 305a is twice the size of the chrominance matrices in both directions. The blocks 310b, 310c from the chrominance matrices 305b, 305c and the associated four blocks 310a from the luminance matrix 305a together form a macroblock 312.

[0023] MPEG-2 uses one of the two picture structures for encoding a video sequence. In the frame structure, all the lines of the video are coded. In the field structure, the two fields of a frame are coded independently of each other, and the top fields and bottom fields are coded in an alternating order. Each of the two fields has its own picture header.

[0024] A picture 305 is divided into slices 315, wherein each slice 315 includes any number of encoded contiguous macroblocks 310 from left to right and top to bottom order. Slices 315 are used in the handling of errors. If a bit stream contains an error, a slice 315 can be skipped allowing better error concealment.

[0025] As noted above, the macroblocks 312 comprise blocks 310 from the chrominance matrices 305b, 305c and the luminance matrix 305a. The blocks 310 are the most basic units of MPEG-2 encoding. Each block 310 from the chrominance matrices 305b, 305c and the associated four blocks from the luminance matrix 305a are encoded, b0.....b5, and together form the data portion of a macroblock 312. The macroblock 312 also includes a number of control parameters including (Coded Block Pattern) CBP 312a, Qscale 312b, motion vector 312c, type 312d, and address increment 312e. The CBP 312a indicates the number of coded blocks in a macroblock. The Qscale 312b indicates the quantization scale. The motion vector 312c is used for temporal encoding. The type 312d indicates the method of coding and content of the macroblock according to the MPEG-2 specification. The address increment 312e indicates the difference between the current macroblock address and the previous macroblock address.

[0026] The macroblocks 312 are encoded using various algorithms. The algorithms take advantage of both spatial redundancy and/or temporal redundancy. The algorithms taking advantage of spatial redundancy utilize discrete cosine transformation (DCT), quantization, and run-length encoding to reduce the amount of data required to code each macroblock 312. Pictures 305 with macroblocks 312 which

are coded using only spatial redundancy are known as Intra Pictures 305I (or I-pictures).

[0027] The algorithms taking advantage of temporal redundancy use motion compensation based prediction. With pictures which are closely related, it is possible to accurately represent or "predict" the data of one picture based on the data of a reference picture, provided the translation is estimated. Pictures 305 can be considered as snapshots in time of moving objects. Therefore, a portion of one picture 305 can be associated with a different portion of another picture 305.

[0028] Pursuant to the MPEG-2 Standard, a macroblock 315 of one picture is predicted by searching macroblocks 315 of reference picture(s) 305. The difference between the macroblocks 315 is the prediction error. The prediction error can be encoded in the DCT domain using a small number of bits for representation. Two-dimensional motion vector(s) represents the vertical and horizontal displacement between the current macroblock 315 and the macroblock(s) 315 of the reference picture(s). Accordingly, the macroblock 315 can be encoded by using the prediction error in the DCT domain at b0.....b5, and the motion vector(s) at 315c describing the displacement of the macroblock(s) of the reference picture(s) 305.

[0029] Pictures 305 with macroblocks 315 coded using temporal redundancy with respect to earlier pictures 305 of the video sequence are known as predicted pictures 305P (or P-pictures). Pictures 305 with macroblocks 315 coded using temporal redundancy with respect to earlier and later pictures 305 of the video sequence are known as bi-directional pictures 305B (or B-pictures).

[0030] Referring now to **FIGURE 2**, there is illustrated a block diagram of an exemplary decoder in accordance with an embodiment of the present invention. Data is output from buffer 532 within SDRAM 530. The data output from the presentation buffer 532 is then passed to a data transport processor 535. The data transport processor 535 demultiplexes the transport stream into packetized elementary stream constituents, and passes the audio transport stream to an audio decoder 560 and the video transport stream to a video transport decoder 540 and then to a MPEG video decoder 545. The audio data is then sent to the output blocks, and the video is sent to a display engine 550. The display engine 550 scales the video picture, renders the graphics, and constructs the complete display. Once the display is ready to be presented, it is passed to a video encoder 555 where it is converted to analog video using an internal digital to analog converter (DAC). The digital audio is converted to analog in an audio digital to analog converter (DAC) 565.

[0031] The decoder also includes a frame buffers 570 for storing frames 305 decoded by the video decoder 545. The frames 305 are stored on a macroblock 312 by macroblock 312 basis into the frame buffers 570.

[0032] The frames 305 are stored in the frame buffers 570 to await display by the display engine 250. Additionally, reference frames 305 are stored in the frame buffer 570 for use in prediction of other frames 305. During both the display and the prediction of another frame 305, the pixels in the frame 105 at specific locations within the frame 305 are retrieved. In order to display a frame 305 or predict another frame 305 from a frame 305 in

real-time, it is advantageous if the address storing particular pixels can be determined from the location within the frame 305 with simple arithmetic operations.

[0033] Referring now to **FIGURE 3**, there is illustrated a block diagram of the frame buffer 570 storing macroblocks 312(0)...312(n) in accordance with another embodiment of the present invention. The frame buffer 570 comprises a plurality of sections 570(0)...570(n). Each section 570(x) comprises 128 consecutive data words 570(x)(0)...570(x)(127). The data words 570(x)(0)...570(x)(127) further comprise 16 bytes for storage of data. Accordingly, each section 570(0)...570(n) comprises 2048 bytes of memory.

[0034] Each section 570(0)...570(n) stores a corresponding set of five macroblocks 312(x)(0)...312(x)(4) therein. As noted above, a macroblock 312 comprises a 16 byte x 16 byte luminance matrix Y (or four 8 byte x 8 byte luminance blocks), and two 8 byte x 8 byte chrominance matrices or blocks, Cr, Cb.

[0035] In each section 570(x), the luminance matrix Y of the first macroblock 312(x)(0) occupies data words 570(x)(0)...570(x)(15), the chrominance matrices Cr and Cb of the first macroblock 312(x)(0) occupy the data words 570(x)(16)...570(x)(23). The chrominance matrix Cr can occupy the first eight bytes of the data words 570(x)(16)...570(x)(23), while the chrominance matrix Cb can occupy the last eight bytes of the data words 570(x)(16)...570(x)(23).

[0036] The second macroblock 312(x)(1) occupies the data words 570(x)(24)...570(x)(47). The luminance matrix Y of

the second macroblock 312(x)(1) occupies data words 570(x)(24)...570(x)(39). The chrominance matrix Cr can occupy the first eight bytes of the data words 570(x)(40)...570(x)(47), while the chrominance matrix Cb can occupy the last eight bytes of the data words 570(x)(40)...570(x)(47).

[0037] The third macroblock 312(x)(2), fourth macroblock 312(x)(3), and fifth macroblock 312(x)(4) are stored in data words 570(48)...570(71), data words 570(72)...570(95), and data words 570(96)...570(119), respectively, in a similar manner as described above. The data words 570(120)...570(127) are unoccupied. The portions of the macroblocks 312(x)(0)...312(x)(4) and the data words 570(x)(0)...570(x)(127) where the macroblocks 312(x)(0)...312(x)(4) are stored are indicated in the following table.

<u>First Macroblock 312(x)(0)</u>	<u>570(0)...570(23)</u>
Luminance Matrix Y	570(0)...570(15)
Chrominance Matrix Cr,Cb	570(16)...570(23)
<u>Second Macroblock 312(x)(1)</u>	<u>570(24)...570(47)</u>
Luminance Matrix Y	570(24)...570(39)
Chrominance Matrix Cr,Cb	570(40)...570(47)
<u>Third Macroblock 312(x)(2)</u>	<u>570(48)...570(71)</u>
Luminance Matrix Y	570(48)...570(63)
Chrominance Matrix Cr,Cb	570(64)...570(71)
<u>Fourth Macroblock 312(x)(3)</u>	<u>570(72)...570(95)</u>
Luminance Matrix Y	570(72)...570(87)
Chrominance Matrix Cr,Cb	570(88)...570(95)
<u>Fifth Macroblock 312(x)(4)</u>	<u>570(96)...570(119)</u>
Luminance Matrix Y	570(96)...570(111)

[0038] Storage of macroblocks 312 in the foregoing manner is advantageous because the macroblocks 312 occupy continuous memory locations, e.g., 24 consecutive data words. The decoder 545 can overwrite a single or up to five macroblocks 312 with new macroblocks 312 from another picture 115, with a single write transaction of the new macroblock 312 to section 570(x).

[0039] Another advantage is that the data word address where each set of five macroblocks 312 begins is offset from the data word address of another set of macroblocks 312 by factor of a power of two, e.g., 128 or 10000000 in binary. Additionally, a frame with a 720 pixel width includes macroblock rows of 45 macroblocks 312. Furthermore, a high portion of the memory locations in the frame buffer 570 are used, e.g., 120 data words/128 data words, or 93.75% utilization.

[0040] Referring now to **FIGURE 4**, there is illustrated a block diagram of the frame buffer 570 storing macroblocks in accordance with another embodiment of the present invention. The frame buffer 570 is a DDR-SDRAM memory and comprises any number of DDR rows 605(0)...605(n), and four banks. Of course, although in the present embodiment, there are four banks, it is noted that alternative embodiment of the present invention may include any number of banks. Each row 605 straddles each of the banks.

[0041] Each row 605 comprises at least 512 Gigantic data words, wherein each gigantic data word comprises 16 bytes. Accordingly, the portion of a row 605 that straddles a particular bank, now referred to as a row bank 610(0,0), 610(0,1), 610(0,2), 610(0,3), ..., 610(n,0), 610(n,1), 610(n,2), 610(n,3) comprises at least 128 data words.

[0042] Each row bank 610 stores a corresponding set of five macroblocks 312(0)...312(4). As noted above, a macroblock 312 comprises a 16x16 luminance matrix Y, and two 8x8 chrominance matrices or blocks, Cr, and Cb. The first set of five macroblocks 312(0)...312(4) are stored in the first 120 data words, data words 0...119, of the row bank 610.

[0043] The portions of the macroblocks 312 and the data words in the row bank 610 where the macroblocks 312 are stored are indicated in the following table.

<u>1st Set Macroblocks 312(0)...312(4)</u>	<u>Data Words 0...119</u>
<u>First Macroblock 312(0)</u>	<u>0...23</u>
Luminance Matrix	0...15
Chrominance Matrix Cr, Cb	16...23
<u>Second Macroblock 312(1)</u>	<u>24...47</u>
Luminance Matrix	24...39
Chrominance Matrix Cr, Cb	40...47
<u>Third Macroblock 312(2)</u>	<u>48...71</u>
Luminance Matrix	48...63
Chrominance Matrix Cr, Cb	64...71
<u>Fourth Macroblock 312(3)</u>	<u>72...95</u>

Luminance Matrix	72...87
Chrominance Matrix Cr, Cb	88...95
<u>Fifth Macroblock 312(4)</u>	<u>96...119</u>
Luminance Matrix	96...111
Chrominance Matrix Cr, Cb	112...119

Each set of macroblocks 312(0)...312(4) can comprise any five macroblocks 312. However, in a particular embodiment, each macroblock set 312(0)...312(4) can comprise five horizontally adjacent macroblocks 312. Additionally, in some cases, it advantageous to prevent vertically neighboring macroblocks 312 from occupying the same row bank 610. Accordingly, the row banks 610 can be populated by sets of macroblocks 312(0)...312(4), pursuant to an algorithm that prevents vertically adjacent macroblocks 312 from occupying the same row bank 610. For example, such an algorithm may first store a section (five horizontally adjacent macroblocks 312(0)...312(4)) or 2 power n sections, in each of the row banks 610 associated with a particular row 605, before storing another section (five horizontally adjacent macroblocks 312(5)...312(9)) or 2 power n sections in the row bank 610.

[0044] The decoder system as described herein may be implemented as a board level product, as a single chip, application specific integrated circuit (ASIC), or with varying levels of the decoder system integrated with other portions of the system as separate components. The degree of integration of the decoder system will primarily be determined by the speed and cost considerations. Because of the sophisticated nature of modern processor, it is

possible to utilize a commercially available processor, which may be implemented external to an ASIC implementation. Alternatively, if the processor is available as an ASIC core or logic block, then the commercially available processor can be implemented as part of an ASIC device wherein certain operations are implemented as instructions in firmware.

[0045] while the invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the invention. In addition, many modifications may be made to adapt particular situation or material to the teachings of the invention without departing from its scope. Therefore, it is intended that the invention not be limited to the particular embodiment(s) disclosed, but that the invention will include all embodiments falling within the scope of the appended claims.